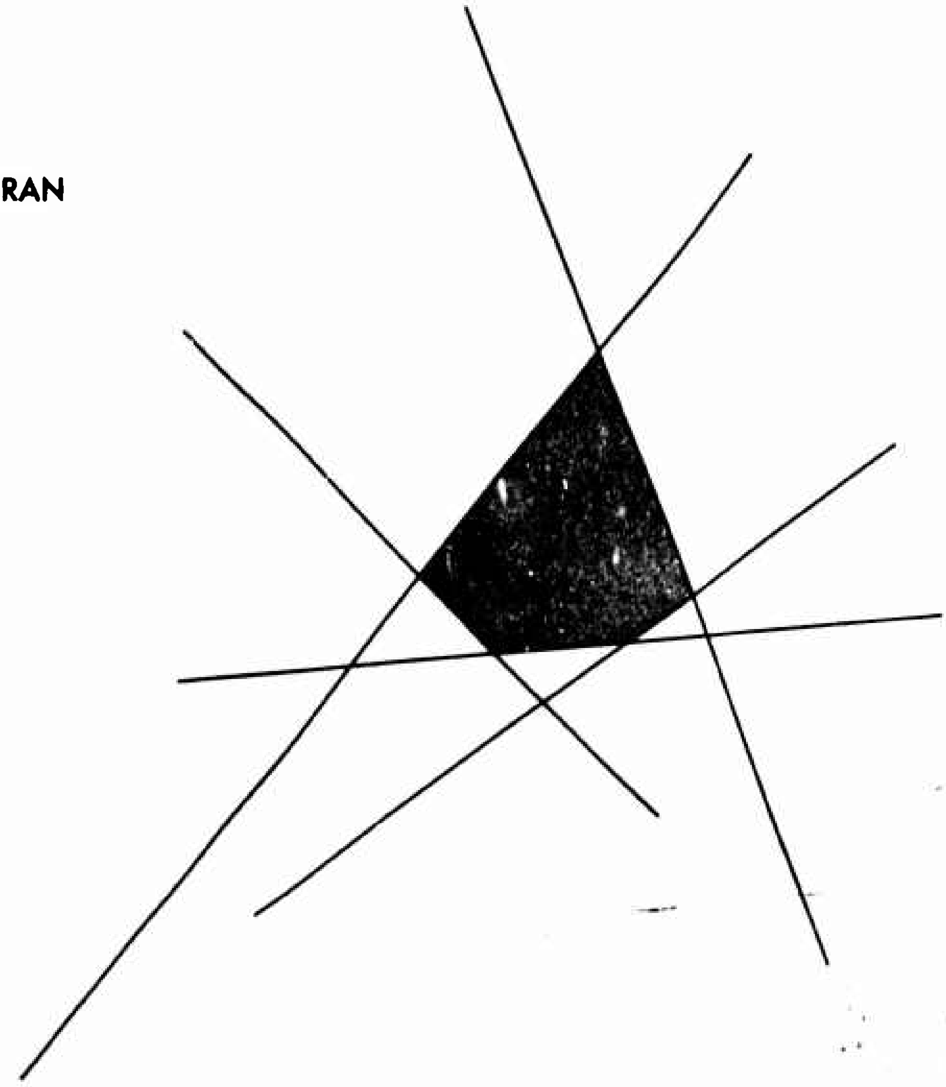
ORC 68-5
MARCH 1968

AN EXPERIMENTAL STUDY ON SOLVING LINEAR PROGRAMS

by

ARUNACHALAM RAVINDRAN



OPERATIONS
RESEARCH
CENTER

UNIVERSITY OF CALIFORNIA · BERKELEY

AN EXPERIMENTAL STUDY ON SOLVING LINEAR PROGRAMS

Arunachalam Ravindran

California University
Berkeley, California

March 1968

AN EXPERIMENTAL STUDY ON SOLVING LINEAR PROGRAMS

by

Arunachalam Ravindran
Operations Research Center
University of California, Berkeley

March 1968 ORC 68-5

This research has been partially supported by the Office of Naval Research under Contract Nonr-222(83), and the National Science Foundation under Grant GP-7417 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

ACKNOWLEDGEMENT

The author takes this opportunity to express his sincere thanks and appreciation to Professor David Gale who was primarily responsible for undertaking this study and whose valuable suggestions and ideas are incorporated in many parts of this paper.

ABSTRACT

An experimental study to compare the simplex method and the Lemke's method to solve linear programs is made. The M3 code for simplex method and the author's code for the Lemke method were used in the study. Comparison was made only with regard to the number of iterations each method takes and our little study shows encouraging results about the superiority of Lemke method, but no general recommendation is made by the author due to size of the study and data. A by-product of our study is a complementary pivot algorithm to solve linear programs which is a modification of the Lemke's method and which saves a considerable storage and time of computation.

AN EXPERIMENTAL STUDY ON SOLVING LINEAR PROGRAMS

by

A. Ravindran

1. INTP ^UCTION

From the beginning of the era of linear programming a lot of work has been done to find a better and more efficient method to solve linear programs than the simplex method of G. B. Dantzig [1] but without much success. Quite a few papers have been published about the performance of the simplex method in terms of the number of iterations or pivot steps. Kuhn and Quandt made a study to test different pivot choices for the simplex code [4]; but strangely, the firm theoretical bounds given in the papers are much higher than the actual number encountered in practice. In spite of extensive computational experience, not very much is known precisely about the performance of the simplex method, and it is still considered to be the best known method to solve linear programs by many workers in the field.

To add to this huge amount of literature on the study of the simplex method, we have made a small scale study to compare it with the "complementary pivotal method" of Lemke and Howson. Our main object has been to compare the number of iterations in both cases. Our study revealed that generally Lemke method is as efficient as simplex and in special class of A matrices (viz $A > 0$) Lemke method takes as little as 1/2 to 1/3 the number of iterations of simplex! This opens the door to further study of the Lemke method on a large scale to solve linear programs. A by-product of our study is a modified algorithm to that of Lemke-Howson to solve linear programs which will save a considerable amount of storage and computation time.

To make this account self-contained, an exposition of the Lemke's method in the form we have used (i.e., programmed[†]) is presented in Section 2. The details of the statistical experiment and the results of our study are described in Section 3. Our modified Lemke's algorithm is explained in Section 4.

The fact that led us to make this study is that the simplex method always starts with a basic feasible solution [1]. In many problems in fact we do not have an initial basic feasible solution readily available and to find one, one uses the so called Phase 1 technique of Dantzig which means solving another linear program with additional artificial variables. This Phase 1 is avoided when one uses Lemke method as it is easy to find one "almost complementary solution."

[†] An independent computer program was written in FORTRAN IV for the Lemke-Howson's algorithm by the author.

2. THE COMPLEMENTARY PROBLEM AND LEMKE'S ITERATIVE TECHNIQUE

The following complementary problem was posed by Dantzig and Cottle in 1963 [2]

$$
\begin{aligned}
w &= Mz + q \\
w &\geqq 0,\ z \geqq 0 \qquad\qquad (1)^{\dagger} \\
w \cdot z &= 0 \quad (\text{i.e., } w_i z_i = 0)
\end{aligned}
$$

where M is (n x n) square matrix.

w,z,q are n-vectors.

It can be seen that linear programming, quadratic programming and bi-matrix (2-person, nonzero-sum) games can be transformed to the above complementary problem form.

Dantzig and Cottle in their paper [2] also proposed an algorithm which is applicable to matrices M that have positive principal minors (in particular to positive definite matrices) and after some modification (which uses the notion of "block pivot") to positive semi-definite matrices. But we will be concerned with the iterative technique of Lemke and Howson [4] for finding equilibrium points of bi-matrix games which was later extended by Lemke [5] to the complementary problem (1). We will essentially give the step by step procedure of the Lemke's algorithm as presented by Gale [3] which is also used as the basis for the computer code written.

Consider the following linear program:

$$
\begin{aligned}
&\text{Minimize } cx \\
&\text{Subject to } Ax \geqq b \qquad\qquad (2) \\
&\qquad\qquad x \geqq 0
\end{aligned}
$$

[†] No special notation to denote the transpose of a vector is used throughout this paper.

Let A be m x n matrix, b is an m-vector, c and x are n-vectors.

The above problem can be transformed into a complementary problem very easily as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 & A \\ -A^T & 0 \end{bmatrix} \begin{bmatrix} y \\ x \end{bmatrix} + \begin{bmatrix} -b \\ c \end{bmatrix} \tag{3}$$

$$\begin{aligned} uy + vx &= 0 \\ x,y,u,v &\geqq 0 \end{aligned} \tag{4}$$

where u-denotes the primal slack variables

y-denotes the dual variables

v-denotes the dual slack variables.

The reader will immediately note that system (3) represents primal and dual feasibility and (4) represents the complementary slackness condition. Comparing these to system (1) will show

$$M = \begin{bmatrix} 0 & A \\ -A^T & 0 \end{bmatrix},\ w = \begin{pmatrix} u \\ v \end{pmatrix},\ z = \begin{pmatrix} y \\ x \end{pmatrix},\ q = \begin{pmatrix} -b \\ c \end{pmatrix} .$$

It should be remarked here that M is a square nonsymmetric positive semi-definite matrix of order m + n .

We shall now describe Lemke's method to solve the original complementary problem (1) which guarantees to terminate with a solution if it exists for the following cases: (i) $M > 0$ (all elements), (ii) M is positive definite, (iii) M having positive principal determinants, (iv) M is positive semi-definite, (v) M co-positive. We shall first start with some definitions. Let $a_1, \dots, a_n$ denote the first n-column vectors of M matrix. $e_1, \dots, e_n$ are the first n unit vectors.

Definition 1: Complementary Solution

A solution $(w,z) \geqq 0$ satisfying $w = Mz + q$ and for each $i = 1, \ldots, n$ either e_i ($\Leftrightarrow$variable w_i) or $-a_i$ ($\Leftrightarrow$variable z_i) is in the basis and not both, i.e., a solution to system (1).

Definition 2: Almost Complementary Solution

A solution $(w,z) \geqq 0$, $Z_o > 0$ to the system $w = Mz + e\,Z_o + q$ where $\underset{(1xn)}{e} = (1, \ldots, 1)$ such that exactly for one i both e_i and $-a_i$ are out of the basis; and of course Z_o is in the basis as $Z_o > 0$.

Lemke's algorithm starts with an almost complementary solution by adding the vector $e\,Z_o$ to $w = Mz + q$ and using a complementary pivot rule (described later) moves from one almost complementary solution to the other until it terminates which is given by the termination conditions described later.

The initial almost complementary solution is found by augmenting the vector $e\,Z_o$ to the system (1) as follows:

$$w - Mz - e\,Z_o = q$$
$$w,z,Z_o \geqq 0$$
$$w \cdot z = 0\ .$$

The initial tableau with respect to the unit basis will be as follows:

Basis Vectors↓	e_1,	e_2,	...,	e_s,	...,	e_n	$-a_1$	......	$-a_n$	$-e$	q
e_1	1						$-m_{11}$	......	$-m_{1n}$	-1	q_1
e_2		1					$-m_{21}$	......	$-m_{2n}$	-1	q_2
$\vdots$										$\vdots$	
e_s				1			$-m_{s1}$	......	$-m_{sn}$	-1	q_s
$\vdots$										$\vdots$	
e_n						1	$-m_{n1}$	......	$-m_{nn}$	-1	q_n

TABLE 1

Note that the initial basis though complementary need not be feasible as some $q_i < 0$.

Step 1:

$-e$ vector (variable Z_o) is brought into the basis so as to replace one of e_i's , namely, e_s from the basis where $q_s = \underset{i=1,2,..,n}{\text{Min}} (q_i)$. If $q_s \geq 0$ then terminate. We have a complementary solution to (1) given by $w_i = q_i \; \forall_i$ and $z_i = 0$. Otherwise perform the pivot as shown below with circled element as pivot element.

The pivot operation yields the following tableau

$$\begin{array}{c|cccccccccccc|c}
 & e_1, & e_2, & \cdots, & e_{s-1}, & e_s, & e_{s+1}, & \cdots, & e_n & -a_1 & \cdots\cdots & a_n & -e & q \\
\hline
e_1 & 1 & & & & -1 & 0 & & 0 & m'_{11} & & m'_{1n} & 0 & q'_1 \\
\vdots & 0 & 1 & & & -1 & \vdots & & \vdots & & & \vdots & & \vdots \\
e_{s-1} & & & & 1 & -1 & 0 & & \vdots & & & \vdots & 0 & \vdots \\
-e & & & & & -1 & 0 & & \vdots & m'_{s1} & & m'_{sn} & 1 & q'_s \\
e_{s+1} & & & & & & 1 & & & & & & 0 & \\
 & & & & & & 0 & & 0 & & & & & \\
e_n & 0 & & & & -1 & 0 & & 1 & m'_{n1} & & m'_{nn} & 0 & q'_n
\end{array}$$

where $$q'_s = -q_s, \; q'_i = q_i - q_s \;\; \forall \, i \neq s$$

$$m'_{sj} = \frac{-m_{sj}}{-1} = m_{sj} \qquad \forall j = 1,2, \ldots, n$$

$$m'_{ij} = -m_{ij} + m_{sj} \qquad \forall j = 1,2, \ldots, n \quad \forall i \neq s \, .$$

Note $$q'_i \geq 0 \;\; \forall i \, .$$

The corresponding basic solution $w_1 = q_1'$, ..., $w_{s-1} = q_{s-1}'$, $Z_o = q_s'$, $w_{s+1} = q_{s+1}'$, ..., $w_n = q_n'$, is an almost complementary solution as both variables w_s and z_s are out of the basis, $Z_o > 0$ and $w, z \geq 0$.

Step 2:

Since both the vectors $-a_s$ and e_s are out of the basis, we bring in $-a_s$ into the basis which will still maintain almost complementarity. To find the vector leaving the basis, form the ratios

$$\frac{q_i'}{m_{is}'} \quad \text{for } i = 1, \ldots, n \text{ and } m_{is}' > 0 \qquad \forall\; i = 1, \ldots, n .$$

Let

$$\frac{q_k'}{m_{ks}'} = \underset{m_{is}'>0}{\text{Min}} \frac{q_i'}{m_{is}'}$$

then w_k leaves the basis. Obtain the new tableau by performing the pivot with a_{ks}' as the pivot element.

Step 3:

Now bring in $-a_k$ into the basis and continue as before until one of the two things happen which indicates termination.

(i) Min ratio happens at the s^{th} row and Z_o leaves the basis. The resulting basic solution (after the pivot) is the complementary solution to (1).

(ii) All m_{ik}' for $i = 1, 2, \ldots, n$ are $\leq 0 \Rightarrow$ there exists no feasible solution to the original problem (1).

In the coding of the algorithm there was a slight modification done. Instead of doing the pivot operation to the whole tableau only the inverse of the basis is kept in the memory and the pivot operations were done only to the inverse. Similar ideas that are common in revised simplex method with explicit inverse are employed.

3. DESIGN, DATA AND RESULTS OF THE EXPERIMENT

In order to evaluate the statistical efficiency of the simplex method and Lemke's method in solving linear programs, a number of linear programs were generated randomly and the solutions computed[†] by both methods. The linear programs generated were of the form: $Ax \geq b$, $x \geq 0$ minimize $c \cdot x$. Elements of matrix A , vectors b and c were chosen randomly from a uniform distribution from 0 and 1. Matrices of size 5 x 5, 10 x 10, 15 x 15 were tried for the A matrix. Initially, 10 problems for each size were generated and for each problem and method the number of iterations and the means of iteration count for each size matrix are noted. They are shown in the following tables.

[†] The calculations were done on the IBM-7094 at the Computer Center, University of California, Berkeley. The M3 code was used for the simplex method and the author's code for the Lemke method.

Results of the Study

SIZE 5 x 5(A > 0)

Prob. No.	No. of Iterations		Comment
	Simplex Method	Lemke Method	
1	14	3	Optimal Sol.
2	10	7	"
3	10	11	"
4	14	7	"
5	11	5	"
6	11	7	"
7	9	5	"
8	12	5	"
9	10	9	"
10	12	15	"
Average	11.3	7.4	$1.52 = \frac{\text{Simplex}}{\text{Lemke}}$

SIZE 10 x 10(A > 0)

Prob. No.	No. of Iterations		Comment
	Simplex	Lemke	
1	25	9	Optimal Sol.
2	28	9	"
3	29	9	"
4	29	15	"
5	24	13	"
6	27	17	"
7	23	13	"
8	33	7	"
9	25	11	"
10	26	15	"
Average	26.9	11.8	$2.27 = \frac{\text{Simplex}}{\text{Lemke}}$

SIZE 15 x 15(A > 0)

Prob. No.	Simplex	Lemke	Comment
1	48	13	Optimal
2	48	15	"
3	52	17	"
4	47	15	"
5	57	13	"
6	43	11	"
7	49	17	"
8	46	23	"
9	45	21	"
10	46	21	"
Average	48.1	16.6	$\frac{48.1}{16.6} = 2.89$

SIZE 15 x 15$\begin{pmatrix} 50\%\text{ of A} > 0 \\ 50\%\text{ of A} < 0 \end{pmatrix}$

Prob. No.	Simplex	Lemke	Comment
1	22	41	Optimal
2	33	27	Optimal
3	22	13	Infeasible
4	30	31	Optimal
5	35	33	Optimal
6	21	18	Infeasible
7	14	17	Infeasible
8	30	31	Infeasible
9	24	27	Infeasible
10	33	35	Infeasible
Average	26.4	27.3	$\frac{26.4}{27.3} = .96$

SIZE (15 x 15) 80% of A = 0
20% of A > 0

	No. of Iterations		
Prob. No.	Simplex	Lemke	Comment
1	24	43	Optimal
2	22	1	Infeasible
3	26	15	Optimal
4	30	19	Optimal
5	36	19	Optimal
6	22	7	Infeasible
7	18	25	Infeasible
8	19	17	Infeasible
9	24	1	Infeasible
10	50	24	Optimal
Average	27.1	17.1	1.58 $= \frac{\text{Simplex}}{\text{Lemke}}$

SIZE (15 x 15) 80% of A = 0
10% of A > 0
10% of A < 0

	No. of Iterations		
Prob. No.	Simplex	Lemke	Comment
1	19	17	Infeasible
2	26	11	Infeasible
3	11	15	Infeasible
4	9	3	Infeasible
5	20	7	Infeasible
6	17	13	Infeasible
7	20	19	Infeasible
8	9	1	Infeasible
9	8	17	Infeasible
10	13	1	Infeasible
Average	15.2	10.4	$\frac{\text{Simplex}}{\text{Lemke}} = 1.46$

Discussion of the Results

The previous tables show when the matrix A was positive, the simplex method takes 1.52 times for 5 x 5, 2.27 times for 10 x 10 and 2.89 times for 15 x 15 more number of iterations compared to Lemke method and hence Lemke method seems more efficient in the number of iterations. The best reason may be that the Lemke method avoids the Phase 1 procedure of simplex method and for $A > 0$ it is possible to get an easy feasible solution to the primal even if we do not use simplex Phase 1. During our discussions with Dantzig [7] it was suggested that we should try some matrices with a lot of zero entries. And even in that excepting 50% positive and negative entries of A matrix, in all the other cases, Lemke method appears to be more efficient even in finding a basic feasible solution compared to Phase 1 procedure. Since the research was directed at one aspect of the methods (viz No. of iterations) and the data are based on the statistics of one class of linear programs, we do not make any conclusive recommendation about the efficiency of either method for solving linear programs. But our little study has evidently shown that the results we have are very encouraging to do a large scale study about these two methods for future investigation.

4. A MODIFIED APPROACH TO SOLVE LINEAR PROGRAMS BY LEMKE'S METHOD

We observed in Section 3 that when the A matrix is strictly positive, it is worthwhile solving the linear program by Lemke method and in other classes of A matrices both methods are equally good. The only question that can be raised is obviously the fact that we are working with a larger size of the matrix (actually twice the size of A matrix when A is square) when using Lemke's method. The larger size will definitely create problems of storage and time of computation. So further investigations were carried out to see whether it is possible to work with a reduced M matrix where $M = \begin{bmatrix} 0 & A \\ -A^T & 0 \end{bmatrix}$.

The first observation was that the "nice" structure of the M matrix is destroyed immediately after the first pivot when -e vector enters the basis (refer to Figures 1 and 2 of Section 2). So the first occurred idea was to take the -e vector to the right-hand side, perturbate it with the q vector and consider Z_o as a "parameter."

The second observation was that if we work with the "perturbed" problem, the "nice" property of M matrix (i.e., the two submatrices below and above main diagonal are skew symmetric) can be retained by rearranging the columns such that the basis columns (identity matrix) appear first. These observations led to the following modified algorithm which works only with the reduced matrix of M which results in a greater saving in storage and time and answers the question raised about the Lemke method at the beginning of the section. We shall present the modified algorithm completely independent of the complementary pivot theory. But those who are familiar with the complementary pivot theory can see that it is nothing but the Lemke method with a slight modification in the pivot rule and is applied only to the reduced M matrix (viz the transpose of A is never carried in the pivots.)

A Complementary Pivot Algorithm

Consider

$$Ax \geqq b$$

$$x \geqq 0$$

$$\text{Min} \quad cx$$

where A is (m x n) matrix c and x are n vectors b is m vector

$$A = \{a_{ij}\} .$$

Let

$u_1, \ldots, u_m$ denote primal slacks

$y_1, \ldots, y_m$ denote dual variables

$v_1, \ldots, v_n$ denote dual slacks

the vectors u and y ($\geqq 0$) are complementary pair of vectors, i.e., in the solution $u_i y_i = 0 \ \forall \ i = 1, \ldots, n$ (u_i and y_i are called complementary variables). Δ^{ly} x and v are complementary in the sense $x_j v_j = 0 \ \forall \ j = 1, \ldots, n$.

Complementary Rule:

When a variable leaves the basis bring the complement of that variable into basis. Define:

$$\underset{\forall i \& j}{a'_{ij} = -a_{ij}} \ ; \ b'_i = -b_i \ \forall \ i .$$

The basic variables are denoted by the symbol "*". The initial tableau is shown below.

The initial basis contains $u_1, \ldots, u_n, v_1, \ldots, v_n$ (primal slacks and dual slacks) as basic variables. In the tableau,

$$d_i = 1 \text{ if } b_i' < 0 (\Leftrightarrow b_i > 0)$$

$$= 0 \text{ otherwise} \qquad i = 1, \dots, m$$

$$c_{n+1} = \dots = c_{n+m} = 0$$

$$f_i = 1 \text{ if } c_i < 0$$

$$i = 1,2, \dots, n, n+1, \dots, n+m$$

$$= 0 \text{ otherwise} .$$

Basic Variables	u_1	......	u_m	x_1	......	x_n	b'	d
* u_1	1			a_{11}'		a_{1n}'	b_1'	d_1
		. . . .						
* u_m			1	a_{m1}'		a_{mn}'	b_m'	d_m
	c_{n+1}	......	c_{n+m}	c_1	......	c_n	c	
	f_{n+1}	......	f_{n+m}	f_1	......	f_n	f	
	y_1	......	y_m	v_1 *	...... *....*	v_n *	Basic Variables	

Step 1:

$$\text{call } M_1 = \underset{d_i>0}{\text{Max}} \left(\frac{-b_i'}{d_i} \right) = \underset{b_i<0}{\text{Max}} \ (-b_i) \quad i = 1, \dots, m$$

$$M_2 = \underset{f_j>0}{\text{Max}} \left(\frac{-c_j}{f_j} \right) = \underset{c_j<0}{\text{Max}} \ (-c_j) \quad j = 1, \dots, n+m$$

$$\text{Let } Z_o(1) = \text{Max } (M_1, M_2) \tag{4.1}$$

Note:

$$(4.1) \Rightarrow b_i' + Z_o(1)d_i \geq 0 \quad \text{and} \quad c_i + f_i Z_o(1) \geq 0$$

If $Z_o(1) \leq 0$ then terminate. Now there are two cases to consider. The max value $Z_o(1)$ may occur

(i) Corresponding to some $j = 1, \ldots, n + m$.

(ii) Corresponding to some $i = 1, \ldots, m$.

Case (i):

$Z_o(1)$ occurs corresponding to $j = 1$ $\left(\text{i.e., } Z_o(1) = \frac{-c_1}{f_1}\right) \Rightarrow v_1$ leaves the basis. (Note: max-ratio cannot occur corresponding to any nonbasic "j" as their $f_j = 0$.) Since a dual variable leaves the basis, we apply the following *dual complementary pivot.*

By our complementary rule, the complementary variable x_1 (look up at the top row of the tableau corresponding to v_1 !) enters basis. (Note x_1 cannot replace v_1 .) By the minimum ratio rule find the basic variable being replaced by x_1 . For this find ratios $\dfrac{b_i' + Z_o(1)d_i}{a_{i1}'}$ for

$$i = 1, \ldots, n \quad \text{and} \quad a_{i1}' > 0 \; . \tag{4.2}$$

Let the minimum of 4.2 occur corresponding to $i = m$. (If no such minimum exists, terminate as we have an unbounded solution (Ray Solution) as $a_{i1}' \leq 0 \;\forall\; i$.) So from min ratio rule, we find x_1 replaces u_m and y_m , the complementary variable of u_m replaces v_1 from the basis.

Using a'_{m1} as pivot element, perform the pivot operation. Note that the new basis is shown again by symbol "*" and this concludes a dual complementary pivot.

		u_1	......	u_{m-1}	u_m	x_1	x_2	......	x_n		
*	u_1	1			$\frac{-a'_{11}}{a'_{m1}}$	0	$\bar{a}_{12}$		$\bar{a}_{1n}$	$\bar{b}_1$	$\bar{d}_1$
			. . . .								
*	u_{m-1}			1	$\frac{-a'_{m-1,1}}{a'_{m1}}$	0	$\bar{a}_{m-1,2}$		$\bar{a}_{m-1,n}$	$\bar{b}_{m-1}$	$\bar{d}_{m-1}$
*	x_1			0	$1/a'_{m1}$	1	$\bar{a}_{m2}$		$\bar{a}_{m,n}$	$\bar{b}_m$	$\bar{d}_m$
		$\bar{c}_{n+1}$		$\bar{c}_{n+m-1}$	$\bar{c}_{n+m}$	0	$\bar{c}_2$		$\bar{c}_n$		
		$\bar{f}_{n+1}$		$\bar{f}_{n+m-1}$	$\bar{f}_{n+m}$	0	$\bar{f}_2$		$\bar{f}_n$		
		y_1			y_m	v_1	v_2		v_n		
					*		*		*		

where

$$\bar{a}_{mj} = \frac{a'_{mj}}{a'_{m1}} \qquad \forall\ j = 2, \ldots, n$$

$$\bar{a}_{ij} = a'_{ij} - a'_{i1}\left(\frac{a'_{mj}}{a'_{m1}}\right) \qquad \forall\ j = 2, \ldots, n \quad \forall\ i = 1, \ldots, m-1$$

$$\bar{b}_m = \frac{b'_m}{a'_{m1}};\ \bar{d}_m = \frac{d_m}{a'_{m1}} \tag{4.3}$$

$$\bar{b}_i = b'_i - a'_{i1}\left(\frac{b'_m}{a'_{m1}}\right) \qquad \forall\ i = 1, \ldots, m-1$$

$$\bar{d}_i = d_i - a'_{i1}\left(\frac{d_m}{a'_{m1}}\right) \qquad \forall\ i = 1, \ldots, m-1$$

$$\bar{c}_j = c_j - c_1\left(\frac{a'_{mj}}{a'_{m1}}\right) \qquad j = 2, \ldots, n + m$$

$$\bar{f}_j = f_j - f_1\left(\frac{a'_{mj}}{a'_{m1}}\right) \qquad j = 2, \ldots, n + m \tag{4.3}$$

(Note: $f_1 = 1$)

Case (ii):

Let max-ratio 4.1 occur corresponding to some $i = 1 \Rightarrow u_1$ leaves the basis. Since a primal variable leaves the basis due to 4.1, we apply *primal complementary pivot* which runs as follows.

By our complementary rule, since u_1 leaves the basis bring y_1 into basis which replaces one of the v_j's $j = 1, \ldots, n$. To find the basic variable being replaced by y_1 apply max-ratio rule as follows: Form the ratios to find

$$\text{Max}\left(\frac{c_j + Z_o(1)f_j}{a'_{1j}}\right) \tag{4.4}$$

$$\forall\ j = 1, \ldots, n \quad \text{and} \quad a'_{1j} < 0$$

(As a matter of fact this should be carried out for all elements of 1st row which are < 0 but in the first pivot step we know the first m elements of Row 1 are ≥ 0 .)

Let the max of 4.4 occur corresponding to variable v_n . (If no max exists, then terminate We have an unbounded solution (Ray Solution as all $a_{1j} \leq 0\ \forall\ j$.) So y_1 replaces v_n and x_n , the complementary variable of v_n , replaces u_1 . Using a'_{1n} as the pivot element perform the pivot operation as in Case (i) (Note: $a'_{m,n} < 0$)

Step 2:

With new tableau at hand, form the ratios as in Step 1 to find

$$\bar{M}_1 = \underset{\bar{d}_i > 0}{\text{Max}} \left(\frac{-\bar{b}_i}{\bar{d}_i} \right) \qquad i = 1, \ldots, m$$

$$\bar{M}_2 = \underset{\bar{f}_j > 0}{\text{Max}} \left(\frac{-\bar{c}_j}{\bar{f}_j} \right) \qquad j = 1, \ldots, n + m$$

$Z_o(2) = \text{Max}[\bar{M}_1, \bar{M}_2]$. If $Z_o(2) \leq 0$, then terminate.

Theorem 4.1:

The new value of $Z_o(2)$ (after a primal or dual complementary pivot) is less than or equal to $Z_o(1)$.

Proof:

(a) Consider first Case (i) where we have a dual complementary pivot:

$$Z_o(1) = \frac{-c_l}{f_l} \;(4.5) \text{ and } f_l = 1$$

and by 4.1,

$$\frac{-c_l}{f_l} \geq \frac{-b'_i}{d_i} \;\forall i$$

and

$$Z_o(1) = \frac{-c_l}{f_l} \geq \frac{-c_j}{f_j} \;\forall j \; . \tag{4.6}$$

To prove $Z_o(2) \leq Z_o(1)$ we have to show

$$\bar{M}_1 \leq Z_o(1) \qquad (4.7)$$

and

$$\bar{M}_2 \leq Z_o(1) . \qquad (4.8)$$

Now

$$\bar{M}_1 = \underset{\bar{d}_i>0}{\text{Max}} \ \frac{-\bar{b}_i}{\bar{d}_i} \ \forall \ i$$

<u>Note</u>: 4.7 and 4.8 => theorem for Case (i). From min-ratio rule (4.2)

$$\frac{b'_m + Z_o(1)d_m}{a'_{m1}} \leq \frac{b'_i + Z_o(1)d_i}{a'_{i1}} \ \forall \ i$$

$$\frac{b'_m}{a'_{m1}} a'_{i1} + \frac{d_m}{a'_{m1}} a'_{i1}Z_o(1) \leq b'_i + Z_o(1)d_i \quad \forall \quad i$$

$$\left(b'_i - \frac{b'_m}{a'_{m1}} a'_{i1}\right) + Z_o(1)\left(d_i - \frac{d_m}{a'_{m1}} a'_{i1}\right) \geq 0 \quad \forall \quad i$$

$$\bar{b}_i + Z_o(1)\bar{d}_i \geq 0 \ \text{ by } (4.3)$$

$$\frac{-\bar{b}_i}{\bar{d}_i} \leq Z_o(1) \quad \forall \quad i \Rightarrow \bar{M}_1 \leq Z_o(1) \Rightarrow 4.7 \ .$$

Similarly

$$\bar{M}_2 = \underset{f_j>0}{\underline{\text{Max}}} \left[\frac{-\bar{c}_j}{\bar{f}_j}\right]$$

we will show

$$\bar{c}_j + \bar{f}_j \; Z_o(1) \geq 0 \; (\Rightarrow 4.8)$$

by 4.3,

$$\bar{c}_j + \bar{f}_j Z_o(1) = \left[c_j - c_1\left(\frac{a'_{mj}}{a'_{m1}}\right)\right] + \left[f_j - f_1\left(\frac{a'_{mj}}{a'_{m1}}\right)\right] Z_o(1) =$$

$$\underbrace{(c_j + Z_o(1)f_j)}_{\substack{\geq 0 \\ \text{by } (4.6)}} - \frac{a'_{mj}}{a'_{m1}} \underbrace{[c_1 + Z_o(1)f_1]}_{\substack{|| \\ 0 \\ \text{by } (4.5)}} \geq 0 \Rightarrow 4.8 \; .$$

Hence after a dual complementary pivot

$$Z_o(2) \leq Z_o(1)$$

(b) Now consider Case (ii) where we have a primal complementary pivot.

$$Z_o(1) = \frac{-b'_1}{d_1} \tag{4.9}$$

and

$$\left.\begin{aligned} &(4.1) \Rightarrow Z_o(1) \geq \frac{-b'_i}{d'_i} && \forall \;\; i \\ &\text{and} \;\; Z_o(1) \geq \frac{-\bar{c}_j}{\bar{f}_j} && \forall \;\; j \; . \end{aligned}\right. \tag{4.10}$$

We will first show:

$$\bar{M}_1 \leq Z_o(1)$$

i.e.,

$$\frac{-\bar{b}_i}{\bar{d}_i} \leq Z_o(1) \qquad \forall \ i$$

$$\bar{b}_i + Z_o(1)\bar{d}_i \geq 0 \qquad \forall \ i$$

$$\text{i.e., } \bar{b}_i + Z_o(1)\bar{d}_i = \left[b'_i - a'_{in}\left(\frac{b'_m}{a'_{ln}}\right)\right] + Z_o(1)\left[d'_i - a'_{in}\left(\frac{d_m}{a'_{ln}}\right)\right] =$$

$$\underbrace{[b'_i + d'_i \ Z_o(1)]}_{\substack{\| \\ \geq 0 \\ \text{by } 4.10}} - \frac{a'_{in}}{a'_{ln}} \underbrace{[b'_l + Z_o(1)d_l]}_{\substack{\| \\ 0 \\ \text{by } 4.9}} \geq 0$$

$$\Rightarrow \ \bar{M}_1 \leq Z_o(1)$$

Q.E.D.

To show:

$$\bar{M}_2 \leq Z_o(1)$$

i.e.,

$$\frac{-\bar{c}_j}{\bar{f}_j} \leq Z_o(1) \qquad \forall \ j \ .$$

By max-ratio rule (4.4)

$$\frac{c_j + Z_o(1)f_j}{a'_{1j}} \leq \frac{c_n + Z_o(1)f_n}{a'_{1n}}$$

(Note: $a'_{1n} < 0$) .

Note:

$$\bar{c}_j = c_j - c_n\left(\frac{a'_{1j}}{a'_{1n}}\right)$$

$$\bar{f}_j = f_j - f_n\left(\frac{a'_{1j}}{a'_{1n}}\right)$$

$$\frac{c_j + Z_o(1)f_j}{-a'_{1j}} \geq \frac{c_n + Z_o(1)f_n}{-a'_{1n}} \quad .$$

Note $-a'_{1n} > 0$ and $-a'_{1j} > 0$

$$\Rightarrow \left[c_j - c_n\left(\frac{a'_{1j}}{a'_{1n}}\right)\right] + \left[f_j - f_n \frac{a'_{1j}}{a'_{1n}}\right] Z_o(1) \geq 0$$

$$\bar{c}_j + \bar{f}_j \; Z_o(1) \geq 0$$

$$\text{i.e.,} \quad \bar{M}_2 \leq Z_o(1)$$

Q.E.D.

Hence the theorem is proved.

After finding $Z_o(2)$, according to the basic variable which leaves the basis either primal or dual complementary pivot is done as in Step 1.

Step n:

Find

$$Z_o(n) \ .$$

Since we did not use any special property of first pivot, similar arguments proves

$$Z_o(n) \leq Z_o(n - 1) \ .$$

Theorem 4.2:

Terminate if $Z_o(n) \leq 0$, and the optimal solutions of both primal and dual are $b_i^{(n)}$ and $c_i^{(n)}$ $\forall$ i .

Proof:

At Step 1, 4.1 ⇒ that

$$b_i' + d_i \, Z_o(1) \geq 0 \qquad \forall \ i$$

$$c_i' + f_i \, Z_o(1) \geq 0 \qquad \forall \ i \ .$$

At Step n - 1, the min-ratio rule (4.2) of the dual complementary pivot, if a dual variable leaves the basis or the max-ratio rule (4.4) of the primal complementary pivot, if a primal variable leaves the basis, guarantees that

$$b_i^{(n)} + d_i^{(n)} \, Z_o(n - 1) \geq 0 \qquad \forall \ i$$

$$c_i^{(n)} + f_i^{(n)} \, Z_o(n - 1) \geq 0 \qquad \forall \ i$$

Also, we have $Z_o(n - 1) > 0$. 4.1 at Step n gives $Z_o(n) \leq 0$ and

$$\Rightarrow b_i^{(n)} + d_i^{(n)} Z_o(n) \geqq 0 \qquad \forall\ i$$

$$c_i^{(n)} + f_i^{(n)} Z_o(n) \geqq 0 \qquad \forall\ i\ .$$

So the above must be true for all values of Z_o between $Z_o(n - 1)$ and $Z_o(n)$ because of linearity. Note $Z_o(n - 1) > 0$ and $Z_o(n) \leqq 0$.

Hence it is true for $Z_o = 0$

$$\Rightarrow b_i^{(n)} \geqq 0 \qquad \forall\ i$$

$$c_i^{(n)} \geqq 0 \qquad \forall\ i\ .$$

Hence the solution we obtain by setting primal basic variables to $b_i^{(n)}$ is feasible.

Claim:

The dual solution obtained by setting

$$y_i = c_{n+i}^{(n)} \qquad i = 1, \ldots, m$$

$$v_j = c_j^{(n)} \qquad j = 1, \ldots, n$$

is feasible for $yA \leqq c$, $y \geqq 0$ which is the dual L.P. Problem to

$$Ax \geqq b$$

$$x \geqq 0$$

$$\text{Min } cx\ .$$

Proof:

Since $c_{n+i}^{(n)} \geqq 0 \Rightarrow y_i \geqq 0$. Note initially at Step $n = 0$

$$c_{n+i}^{(0)} = 0\ .$$

So y_i's can be considered as multipliers corresponding to the rows of $-A$

matrix which gives $c_j^{(n)}$ $i = 1, \ldots, n$, i.e.,

$$c_j^{(n)} = c_j^{(0)} + y(-a_j) \qquad \forall \; j$$

where a_j is jth column of A . Since

$$c_j^{(0)} = c_j$$

and

$$c_j^{(n)} \geqq 0$$

$\Rightarrow$ that $c_j - ya_j \geqq 0 \;\; \forall \; j$ or $ya_j \leqq c_j \;\; \forall \; j \Rightarrow y$ is feasible for dual.

Q.E.D. (Claim)

Note our primal and dual complementary pivot rule maintains the complementarity between primal and dual variables, i.e., either y_i or u_i is in the basis but not both. Similarly either x_i or v_i is in the basis but not both. Hence the complementary slackness is also satisfied by the primal and dual solution got from $b_i^{(n)}$ and $c_i^{(n)}$. Hence they are optimal since by claim they are feasible.

Finiteness of the Algorithm:

Under nondegeneracy assumption (i.e., no ties when forming max-ratio or min-ratio rule) $Z_o(n)$ decreases with n (proved already), and since we terminate when $Z_o(n) \leqq 0$ the algorithm clearly ends in finite number of steps.

Comments:

Those who are familiar with the complementary pivot theory, can easily see that in our algorithm we are essentially applying the Lemke's method but to a reduced tableau, taking advantage of the structure of M-matrix.

Dantzig [1] has claimed that the Lemke's complementary pivot method to solve linear programs is indentical with respect to the pivot steps to his self-dual parametric algorithm [1]. Though this fact is not very obvious to see, using our algorithm we can see that the pivot steps are identical to those in the self-dual parametric algorithm though Dantzig uses the idea of primal and dual simplex method while we make use of the complementarity between the variables. Since our method is a condensed form of Lemke's method in some sense, we have also shown that Dantzig's claim is true.

Illustration:

Consider the following linear program: [1]

$$\begin{array}{ll} \text{Minimize} & 3x_1 - 3x_2 \\ \text{Subject to} & -2x_1 - 2x_2 \geq -10 \\ & x_1 - x_2 \geq 1 \\ & -x_1 + 2x_2 \geq 1 \\ & x_1 , x_2 \geq 0 \end{array}$$

Using the algorithm of Section 4, we start with the following initial tableau.

basis	u_1	u_2	u_3	x_1	x_2	b'	d
* u_1	1	0	0	2	2	10	0
* u_2	0	1	0	-1	$\textcircled{1}$	-1	1
* u_3	0	0	1	1	-2	-1	1
	0	0	0	3	-3	$\underset{\sim}{c}$	
	0	0	0	0	1	$\underset{\sim}{f}$	
	y_1	y_2	y_3	$\underset{*}{v}_1$	$\underset{*}{v}_2$		

Basic variables are designated by the symbol "*"

<u>Step 1</u>

$$M_1 = \max_{d_i > 0} \{+1, +1\} = 1$$

$$M_2 = \max_{f_j > 0} \{3\} = 3$$

$Z_o(1) = 3$ and => v_2 a dual variable leaves the basis. So we apply dual complementary pivot rule. The complementary virable (looking up the tableau corresponding to v_2) x_2 enters the basis and replaces u_2 by the min. ratio rule as $\min_{a'_{i5}>0} \left\{ \frac{10}{2}, \frac{-1+1}{1} \right\} = 1$. Hence y_2, the complementary variable of u_2 replaces v_2 from the basis. The new tableau after the pivot operation is as shown below

	u_1	u_2	u_3	x_1	x_2	$\bar{b}$	d
* u_1	1	-2	0	4	0	12	-2
* x_2	0	1	0	-1	1	-1	1
* u_3	0	2	1	$\boxed{-1}$	0	3	3
	0	3	0	0	0	c	
	0	-1	0	1	0	f	
	y_1	$\underset{*}{y_2}$	y_3	$\underset{*}{v_1}$	v_2		

<u>Step 2</u>:

$$M_1 = \text{Max } \{1, 1\} = 1$$

$$M_2 = \text{Max } \{0\} = 0$$

$Z_o(2) = 1$ since we have ties here we just pick one, say, u_3 . So u_3 , a primal variable leaves the basis and we use primal complementary pivot rule by which y_3 enters the basis and replaces by max ratio rule v_1 as $\max\limits_{\bar{a}_{3j}<0} \left\{ \frac{0 + 1}{-1} \right\} = -1$ so so x_1 replaces u_3 . The pivot element is circled. The new tableau after the pivot is as shown below:

	u_1	u_2	u_3	x_1	x_2	$\bar{b}$.	d ↓
* u_1	1	6	4	0	0	0	10
* x_2	0	-1	-1	0	1	2	-2
* x_1	0	-2	-1	1	0	3	-3
	0	3	0	0	0	$\underset{\sim}{c}$	
	0	1	1	0	0	$\underset{\sim}{f}$	
	y_1	$\underset{*}{y_2}$	$\underset{*}{y_3}$	v_1	v_2		

Step 3:

$$M_1 = \left\{\frac{-0}{10}\right\} = 0$$

$$M_2 = \text{Max}\left\{\frac{-3}{1}, \frac{-0}{1}\right\} = 0$$

$Z_o(3) = 0$ terminate optimal basis is shown by * and their values by looking up at the corresponding b and c Rows, i.e.,

$u_1 = 0$	and	$y_2 = 3$
$x_2 = 2$		$y_3 = 0$
$x_1 = 3$		
(Primal basis)		(Dual basis)
The others		The others
$x_3 = 0$		$y_1 = 0$
$u_2 = 0$		$v_1 = 0$
		$v_2 = 0$

Minimum = 3.3 - 3.2 = 9-6 = $\underline{\underline{3}}$

REFERENCES

[1] Dantzig, G. B., LINEAR PROGRAMMING AND EXTENSIONS, Princeton University Press, Princeton, New Jersey, (1963).

[2] Dantzig, G. B. and R. W. Cottle, "Positive (Semi) Definite Programming," ORC 63-18, Operations Research Center, University of California, Berkeley, (May 1963).

[3] Gale, David, Lecture Notes of Mathematical Programming Course, University of California, Berkeley, (Spring 1967).

[4] Kuhn, H. W. and R. E. Quandt, "An Experimental Study of the Simplex Method," PROCEEDINGS IN SYMPOSIA IN APPLIED MATHEMATICS, Vol. XV, pp. 107-124.

[5] Lemke, C. E. and J. T. Howson, Jr., "Equilibrium Points of Bi-Matrix Games," J.Soc.Indust.Appl.Math., Vol. 12, pp. 413-423, (1964).

[6] Lemke, C. E., "Bimatrix Equilibrium Points and Mathematical Programming," Management Science, Vol. 11, pp. 681-689, (1965).

[7] Dantzig, G. B., Private Discussions.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) University of California, Berkeley	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

AN EXPERIMENTAL STUDY ON SOLVING LINEAR PROGRAMS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Research Report

5. AUTHOR(S) (Last name, first name, initial)

RAVINDRAN, Arunachalam

6. REPORT DATE March 1968	7a. TOTAL NO. OF PAGES 31	7b. NO. OF REFS 7

8a. CONTRACT OR GRANT NO. Nonr-222(83) b. PROJECT NO. NR 047 033 c. Research Project No.: RR 003 07 01 d.	9a. ORIGINATOR'S REPORT NUMBER(S) ORC 68-5 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES Also supported by the National Science Foundation under Grant GP-7417.	12. SPONSORING MILITARY ACTIVITY MATHEMATICAL SCIENCE DIVISION

13. ABSTRACT

An experimental study to compare the simplex method and the Lemke's method to solve linear programs is made. The M3 code for simplex method and the author's code for the Lemke method were used in the study. Comparison was made only with regard to the number of iterations each method takes and our little study shows encouraging results about the superiority of Lemke method, but no general recommendation is made by the author due to size of the study and data. A by-product of our study is a complementary pivot algorithm to solve linear programs which is a modification of the Lemke's method and which saves a considerable storage and time of computation.

Unclassified

Security Classification

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Linear Programs						
Complementary Problem						
Iterations						
Primal Complementary Pivot						
Dual Complementary Pivot						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through

_______________________________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through

_______________________________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through

_______________________________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.

Unclassified

Security Classification